#13
3-19-04

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Tinku Acharya et al.	§	Group Art Unit:	2625
Serial No.:	09/723,123	§		
		§	Examiner:	Timothy M. Johnson
Filed:	November 27, 2000	§		
For:	Wavelet Zerotree Image Coding of Ordered Bits	§	Atty. Dkt. No.:	ITL.0210P1US (P7057X)
Customer No.:	21906	§	Confirmation No.:	5940

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

Applicants respectfully appeal from the final rejection mailed December 29, 2003.

I. REAL PARTY IN INTEREST

The real party in interest is the assignee Intel Corporation.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF THE CLAIMS

Claims 1-15 have been finally rejected and are the subject of this appeal.

Date of Deposit: March 9, 2004
I hereby certify under 37 CFR 1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated above and is addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450.

Rebecca R. Ginn

IV. STATUS OF AMENDMENTS

All amendments are believed to have been entered.

V. SUMMARY OF THE INVENTION

Referring to Figure 2, an embodiment 119 of a compression program, in accordance with one embodiment of the invention, may enable a processor 112 to encode wavelet coefficients in a bit-wise fashion in a technique which may be called modified embedded zerotree (MEZT) coding. In this manner, instead of classifying the wavelet coefficients (as zerotree roots or isolated zeros, as examples), the processor 112 may produce codes to classify the bits of the wavelet coefficients. For example, in some embodiments, the processor 112 may classify a particular bit as being either a zerotree root, an isolated zero, a positive node or a negative node. Unlike conventional zerotree coding schemes, thresholds are not computed to identify insignificant values, as the "0" bit is treated as being insignificant and the "-1" and "1" bits are treated as being significant. Specification, at page 3, line 23 through page 5, line 2.

In this manner, the processor 112 may generate one of the following codes to classify a particular bit: a "P" code to indicate a positive node if the bit indicates a "1"; an "N" code to indicate a negative node if the bit indicates a "-1"; an "R" code to indicate that a "0" bit is a zerotree root; and an "IZ" code to indicate that a "0" bit is an isolated zero. In some embodiments, a particular bit is classified as a negative node only if the bit is the most significant nonzero bit and the bit indicates a "-1." For example, for a coefficient of "-3" that is represented by the three bits "-011," the processor 112 generates an N code to represent the middle bit. However, for this example, the processor 112 generates a P code to represent the least significant bit. Specification, at page 5, line 3 through line 11.

For purposes of providing the wavelet coefficients, the processor 112 may, via wavelet transformations, decompose coefficients that represent pixel intensities of an original image. These wavelet coefficients, in turn, form subbands that are located in multiple decomposition levels. To classify the bits, the processor 112, in some embodiments, may execute the program 119 to process the bits based on their associated bit position, or order. In this manner, the bits of each bit order form a hierarchical tree that the processor 112 may traverse to classify each of the

bits of the tree as being either a zerotree root, an isolated zero, a negative node or a positive node. Thus, as an example, the most significant bits of the wavelet coefficients(this bit may also be zero) are associated with one hierarchical tree (and one bit order), and the next most significant bits are associated with another hierarchical tree (and another bit order). Specification, at page 5, line 12 through line 22.

In some embodiments, the processor 112 indicates the P, N, IZ and R codes via a bit stream that progressively indicates a more refined (i.e., a higher resolution) version of the original image over time. For example, the processor 112 may use the bits "00" to indicate the "P" code, the bits "01" to indicate the "N" code, the bits "10" to indicate the "R" code and the bits "11" to indicate the IZ code. Other coding schemes are possible. The progressive nature of the bit stream is attributable to the order in which the processor 112 processes the bit orders. For example, in some embodiments, the processor 112 may process the bit orders in a most significant first fashion. Therefore, the processor 112 may initially produce code for all the bits that have the highest bit order, then produce code for all of the bits that have the next highest bit order, etc. As a result of this progressing coding, the resultant bit stream may initially indicate a coarser version of the original image. However, more refinements to the image are indicated by the bit stream over time, as the processor 112 produces the codes for the bits having the lower bit orders. Thus, in some embodiments, the resolution of the image that is indicated by the bit stream improves over time, a feature that may be desirable for bandwidth-limited systems. As a result, a decrease in resolution of the reconstructed image may be traded for a decrease in communication bandwidth. Specification, at page 5, line 28 through page 6, line 13.

Referring to Figure 3, in some embodiments, the processor 112 process the bits of each order in a predefined sequence. For example, for a particular bit order, the processor 112 may begin with the highest decomposition level and produce codes for the bits of the highest decomposition level before proceeding to produce codes for the bits of the next highest decomposition level. The processor 112 produces code(s) for the bit(s) of the LL subband and, then for each decomposition level, produces code(s) for the bit(s) of the LH subband, subsequently, produces code(s) for the bit(s) of the HL subband and lastly, produces code(s) the bit(s) of the HH subband. Specification, at page 6, line 14 through line 21.

As an example, the wavelet coefficients produced by a two level decomposition may be arranged in a matrix 40 that is depicted in Figure 4. In this manner, the matrix 40 may be viewed as being subdivided into four quadrants 30a, 30b, 30c and 30d. The upper right 30b, lower left 30c and lower right 30d quadrants includes the coefficients for the LH, HL and HH subband images, respectively, of the first decomposition level. The coefficients for the LL, LH, HL and HH subband images of the second decomposition level are located in the upper left 32a, upper right 32b, lower left 32c and lower right 32d quadrants of the upper left quadrant 30a. The coefficients produced by further decomposition may be arranged in a similar manner. For example, for a third level of decomposition, the upper left quadrant 32a includes the wavelet coefficients of the LL, LH, HL and HH subbands of the third decomposition level. Specification, at page 6, line 22 through page 7, line 2.

As depicted in Figure 5, for the first decomposition level, the coefficients for the LH, HL and HH subband images are represented by the following respective matrices:

$$\begin{bmatrix} E_1 & E_2 \\ E_3 & E_4 \end{bmatrix}, \begin{bmatrix} F_1 & F_2 \\ F_3 & F_4 \end{bmatrix}, \begin{bmatrix} G_1 & G_2 \\ G_3 & G_4 \end{bmatrix}$$

It is noted that each coefficient of the second decomposition level (except A), is associated with at least four coefficients of the first decomposition level, i.e., each coefficient of the first decomposition level has at least four descendant coefficients in the second decomposition level. Therefore, each bit in the first decomposition level has at least four descendent coefficients in the second decomposition level. Specification, at page 7, line 3 through line 16.

For each bit order, the processor 112 may process the bits in the scanning sequence described above. If a particular bit indicates a "1" or a "-1," then the processor 112 generates the P or N code and proceeds to process the next bit in the scanning sequence. However, if a particular bit indicates a "0," then the processor 112 may trace the bit through its descendants to determine if the bit is an isolated zero or a zerotree root. The coefficients in the LL subband are simply entropy encoded. Specification, at page 7, line 18 through line 22.

As an example, to produce the code for the least significant bit (called D(1)) of the D coefficient (located in the HH subband of the second decomposition level), the processor 112 determines whether the D(1) bit indicates a "0." If so, the processor 112 evaluates the descendant bits G1(1), G2(1), G3(1) and G4(1) of the subband HH of the first decomposition

level in search of a "1" or "-1," as indicated in Figure 6. If one of these bits indicates a "1" or "-1," then the D(1) bit is an isolated zero. Otherwise the D(1) bit is a zerotree root. Specification, at page 7, line 23 through line 29.

Therefore, the processor 112 begins the encoding by generating codes for the third order bits (i.e., the most significant bits, which may be zero also) of the coefficients. More particularly, to generate the codes for the third order bits, the processor 112 follows the path 28 (see Figure 5) and produces the appropriate code for the third bit of each coefficient along the path 28. If a particular bit indicates a "0," then the processor 112 evaluates the descendents of the bit to find isolated zeros and zero roots. The coding of the third order bits by the processor 112 produces the following codes (listed in the order of production): P,R,R,R. Subsequently, the processor 112 produces the codes for the second order bits (listed in order of production): IZ,IZ,N,R,IZ,P,IZ,IZ,IZ,P,IZ,IZ. Lastly, the processor 112 produces the codes for the first order bits (listed in order of production): IZ,P,IZ,R,P,IZ,IZ,P,IZ,P,IZ,P. As described above, the processor 112 may indicate the codes via a two bit coding scheme and transmit the codes as produced via a bit stream. Specification, at page 8, line 1 through line 19.

Referring to Figure 7, to summarize, the compression program 119, when executed by the processor 112 may cause the processor 112 to perform the following procedure to produce the above-described coding. First, the processor 112 may express (block 72) a matrix of decomposed coefficients in a signed binary representation. Next, the processor 112 may determine (block 74) the number of digits that are needed to represent the absolute value of the maximum wavelet coefficient. This processor 112 uses a variable (called n) that indicates the current bit order being processed by the processor 112. In this manner, the processor 112 uses a software loop to process the bits, one bit order at a time. To accomplish this, the processor 112 produces codes (block 76) for the bits of the current bit order the using the techniques described above. Subsequently, the processor 112 determines (diamond 78) whether the rate of transmitted bits may exceed a predetermined bit rate. If so, the processor 112 terminates the coding for the current image to comply with the predetermined bit rate. Otherwise, the processor 112 determines (diamond 80) if all bit orders have been processed, i.e., the processor 112 determines if n equals "1." If not, the processor 112 decrements (block 75) the order that is indicated by the n variable by one and proceeds to block 76 to traverse the loop another time to produce codes for

the bits of another bit order. Otherwise, the coding is complete. Specification, at page 9, line 14 through page 10, line 7.

The codec 800 input 802, shown in Figure 8, includes frames of the incoming video sequence. The frames are coded by the codec 800 as intra (I), predicted (P) or skipped (S) frames. The I frame, which is sent at regular intervals starting from the first frame, contains the result of arithmetic coding (AC) 824 on the modified embedded zerotree (MEZT) coded 826 discrete wavelet transformed (DWT) image 828, as described previously herein. Specification, at page 11, line 15 through line 22.

An error image, error frame or error data is the difference between two frames of image data. For the first frame, which is an intra or I -frame, MEZT may be applied as indicated at 826. The next frame is the predicted or P-frame, that is not directly encoded. Instead, the difference from the first or I frame is determined at 808 and that difference is encoded at 812. Specification, at page 11, line 23 through line 27.

The reconstructed predicted frame is the result of encoding error frames using reverse embedded zerotree coded error frames REZT 812 followed by inverse REZT (IREZT) 840. REZT will be explained later. A reconstructed image 830 is then developed from the motion estimation 804 to develop the skipped or S frame 832 and the predicted or P frame 806. The reconstructed frame 830 goes to the inverting input of summer 808. The error compensation 816 is developed from the error frame from IREZT 840 and added to the S frame at 832 to get the P frame 806. Specification, at page 11, line 28 through page 12, line 3.

When the error accumulated by this process crosses a certain threshold as determined at 830, the codec 800 is partially refreshed by transmitting the stream generated by application of AC 812 on a REZT 812 error frame along with the motion vectors 832. Such frames, denoted by P, may be sent at an optimal frequency to maintain a high CR as well as a high peak signal to noise ratio (PSNR) of the reconstructed sequence. Specification, at page 12, line 22 through line 27.

Thus, switching between I, P and S frames is controlled by the energy of the error frame. The performance of the codec 800 may be primarily dependent on the efficiency of REZT 810 and hence also on the correctness and efficiency of MRME 804. Specification, at page 12, line 28 through line 30.

For the other frames, the ith frame is predicted from the previously reconstructed frames ((i-1)th) 920 and the transmitted motion vectors 922. In case of the P frame, the error frame 912 is added to the predicted frame 914 to complete the reconstruction process. The S frame is the motion vector without the error frame. Finally, inverse DWT (IDWT) 916 is applied to each of the frames to get the reconstructed sequence. Specification, at page 13, line 11 through line 15.

In one embodiment, the first block 828 of the codec 800 does the DWT operation, as shown in Figure 8. DWT results in decomposition of each of the input frames into a multi-resolution subband structure. Unlike the discrete cosine transform (DCT), discrete wavelet transformed images contain a lower resolution version of the original image that is usually called the low-frequency subband. Parameters such as the filter coefficients, the number of decomposition levels, etc. can be chosen depending upon the image sequence and its intended application. Moreover, for two-dimensional DWT, filtering is actually applied separately along each dimension, which makes it parallelly realizable and hence suitable for real time applications. Specification, at page 13, line 16 through line 24.

In case of still image compression (i.e., the I frames where no prediction is applied), the matrix obtained after DWT is coded using MEZT scheme as indicated at block 826. This is an efficient bit-plane wise embedded zerotree coding scheme. Specification, at page 13, line 25 through line 27.

Wavelet transform decomposes a video frame into a set of sub-frames with different resolutions corresponding to different frequency bands. These multi-resolution frames provide a representation of the global motion vectors of the video sequence at different scales. Specification, at page 14, line 3 through line 6.

In one embodiment, the motion estimation 804 may be carried out using a three step search algorithm. But in principle, any other motion estimation technique may be suitable for applying in multiresolution hierarchical DWT subbands. The subsampled image (low-frequency subband) is first broken into blocks. Then for each block of a current frame (the frame for which motion estimation is being carried out) a matching block from a previous frame is identified using a distance criterion. The distance criterion may be a minimum mean square error, minimum mean of pixel by pixel absolute differences, or maximum matching pixel count as a few examples. Sum of Pixels by Pixel Absolute Difference (SAD) may be used as the distance

criterion for choosing the best match. The frame is reconstructed at the transmitter end from the previous frame and the motion vectors for purposes of comparison. Specification, at page 14, line 14 through line 24.

In a three step search, SAD is calculated, at the center and at eight specific points within search window in the first step. The distances of these positions are four pixels away from center of the block of interest. The positions are the eight neighbors of a pixel. Depending upon the values of SAD at each search position, the next step search is carried out. At the next step, the search positions are along the same directions but around the position where minimum SAD was found at the previous step and the distance is reduced to two pixels instead of four pixels. The minimum SAD position is found in this step. The last step search is carried out around this position and now the distance is only one pixel. Here the window size becomes seven pixels (4+2+1) along the x or y direction. So the search window is 15 X 15 pixels. Specification, at page 14, line 25 through page 15 line 3.

As the multi-resolution motion estimation approach may drastically reduce the computation time, more computationally involved (but better) search strategies may be used. In the extreme case, a full search may be used. Here, a search window is located around the current block and SAD is calculated for all positions within that search window. A full search may be performed as the search space is small and a better estimate of the block motions results in less prediction error, thus reducing the bit budget. Specification, at page 15, line 7 through line 12.

In one embodiment, the block LL may be transmitted as it is. However, motion estimation or compensation may also be implemented on this block, as it is the same image at a lower resolution. For the block HL motion prediction may be performed. The block size taken was 8*8 and the search area was restricted to a small value in this example. For the LH block the motion estimation may be done over a reduced search area by using the motion vectors calculated for the corresponding blocks in the HL subblock as an initial guess. For the HH block no motion prediction may be done, since a visual system is less sensitive to changes along the diagonal direction. Specification, at page 15, line 13 through line 20.

For decoding, the block LL may be stored without any modification in one embodiment. For the blocks HL and LH the values are reconstructed using the associated motion vectors and the previous frame. For the block HH, the average of the motion vectors for the corresponding

HL and LH positions may be used to reconstruct from the previous frame. Specification, at page 15, line 21 through line 25.

The DWT-based video codec 800 may be computationally efficient because of reduced computational requirements in the multi-resolution motion estimation and bit-plane wise embedded zerotree coding schemes both for DWT frames and the error frames after motion estimation and compensation. The encoding scheme may work in one as opposed to two passes. This makes the codec 800 suitable for implementation both in software and hardware. Specification, at page 16, line 4 through line 9.

The REZT embedded coding scheme is suitable for error images or frames. The error image can be generated by taking difference of two successive DWT images in an image sequence. In the context of video, the error frame is the difference of the original frame from the reconstructed previous frame in the DWT domain. Motion prediction followed by motion compensation leads to generation of reconstructed frames. In error frames, the efficiency is increased by applying the embedded coding in HL, LH, and HH subbands only as shown in Figure 10. The LL subband can be transmitted without any change. The embedded coding may be performed in every bit-plane in one embodiment. Specification, at page 16, line 10 through line 17.

The scheme may be computationally faster than the classical EZT technique. This scheme successfully avoids passing over the smaller or insignificant coefficients in every pass by encoding them in course of the initial passes. Further, the two passes of the classical scheme has been clubbed into a single pass. Specification, at page 23, line 8 through line 11.

Further compression can be achieved when the initial passes(1,2,..) are dropped progressively. The decoding scheme remains same as stated earlier with a minor modification: suitable number of zeros (depending on the number of passes dropped) are to be appended to the decoded data. However, such a process results in PSNR loss. Specification, at page 23, line 12 through line 15.

VI. ISSUES

A. Is claim 1 anticipated by the Zador reference ?

B. Is claim 2 rendered obvious over the Zador reference and in view of the Sodagar reference ?

C. Is claim 5 rendered obvious over the Zador reference and in view of the Ogata reference ?

VII. GROUPING OF THE CLAIMS

For purposes of this appeal, claims 1, 3-4, 6-15 may be grouped together. The patentability of this group and claims 2 and 5 is discussed below.

VIII. ARGUMENT

A. Is claim 1 anticipated by the Zador reference ?

The method of claim 1 includes representing error data as a collection of ordered bits, coding the bits of each order to indicate zerotree roots that are associated with the order, performing wavelet transformations on the image with error data to provide wavelet coefficients for a wavelet transformed error image, and in a single pass, embedding zerotree coding of the wavelet transformed error image while encoding insignificant wavelet coefficients in the course of initial passes.

1. Wavelet transformations on an image with error data to provide wavelet coefficients for a wavelet transformed error image must be taught.

The rejection fails to show that Zador teaches or suggests, performing wavelet transformations on the image with error data to provide wavelet coefficients in a wavelet transformed image. Instead, the Zador reference in column 8, line 60 simply teaches transforming the image data from the original image space to a desired transformed space with

an appropriate wavelet filter transform. In addition, application of a wavelet transform through a known image is indicated in column 13 of the Zador reference.

The Zador reference teaches a modified zero-tree coding method in which a digital image may be compressed to obtain a compressed image data set for subsequent reconstruction. While compressing a digital image, a wavelet decomposition upon color planes of the image is performed. There is no teaching whatsoever as to performing wavelet transformations on the image with error data. In this manner, no wavelet coefficients are provided for a wavelet transformed error image. Nowhere does the Zador reference performs wavelet transformations on the image with error data to provide wavelet coefficients for a wavelet transformed error image.

2. In a single pass, embedding of zerotree coding of the wavelet transformed error image must occur while encoding insignificant wavelet coefficients in the course of initial passes.

The Zador reference fails to teach or suggest embedding zerotree coding of wavelet transformed error image in a single pass while encoding insignificant wavelet coefficients in the course of initial passes. Therefore, the combination of performing wavelet transformation and embedding zerotree coding is not taught or even suggested by the Zador reference. That is, nowhere in the Zador reference a single pass embedding zerotree encoding of wavelet transformed error image is taught or otherwise suggested.

The Zador reference teaches use of a single pass zero-tree in which encoding of the insignificant wavelet coefficients do not occur in the course of initial passes. That is, the Zador reference merely teaches a single pass zero-tree technique without encoding insignificant wavelet coefficients in the course of initial passes. *See,* col. 24, ll. 28-46 in the Zador reference.

In this manner, a compressed image data set is obtained by coding the resulting data set with a lossless entropy encoder. However, there is no teaching as to embedding zerotree coding of the wavelet transformed error image while encoding insignificant wavelet coefficients in the course of initial passes.

3. Representation of error data as a collection of ordered bits must be taught.

Use of error data is not taught or suggested by the Zador reference, as claimed in claim 1. In this manner representation of error data as a collection of ordered bits cannot be taught or suggested by the Zador reference. The Zador reference teaches that a resolution of 16 bits for coefficients in color conversion stage and 12 bits per image color plane in signal processing stage is needed. However, no representation of error data as a collection of ordered bits is taught or suggested in the Zador reference.

Instead of coding the bits of each order to indicate zerotree roots that are associated with the order, the Zador reference simply teaches a scheme for labeling zero-tree roots, for example, in column 23, lines 11-22 in the Zador reference. Accordingly, neither the representation of error data as a collection of ordered bits nor coding of bits of each order to indicate zerotree roots that are associated with the order is taught by the Zador reference. Absent such teachings, all the claim limitations of claim 1 are not anticipated by the Zador reference.

In the advisory action, the Examiner reasons that both the indication of zerotree roots as well as wavelet transformation of error data (e.g., video) is widely known, as is a single pass algorithm used in zerotree coding. However, no specific citation or reference is provided to indicate such teaching. Absent a specific hint or a teaching, claim 1 is not anticipated in view of the Zador reference.

The Zador reference does not provide wavelet coefficient for a wavelet transformed error image because wavelet transformations on the image with error data is not indicated in the Zador reference. Likewise, encoding of the insignificant wavelet coefficients in the course of initial passes is not provided in the Zador reference because the Zador reference does not embed zerotree coding of the wavelet transformed error image in a single pass. Instead, the Zador reference in column 4 line 30 teaches that after wavelet decomposition of the image the convolution pyramid is obtained.

Therefore, consideration of specific claim limitations of independent claim 1 is respectfully requested. As such, the Zador reference fails to use error data as indicated in claim 1 and does not perform wavelet transformations and embed zerotree coding as indicated in the claim 1 limitations. Absent a specific teaching or a hint which entail these specific limitations, all the claim limitations in claim 1 are not anticipated by the Zador reference. Accordingly, the Section 102 rejection of independent claim 1 should be reversed.

B. Is claim 2 rendered obvious over the Zador reference and in view of the Sodagar reference ?

Claim 2 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over the Zador reference and in view of U.S. Patent No. 6,157,746 to Sodagar, et al. (hereinafter, "Sodagar"). Claim 2 calls for determining which of the bits indicate zeros and classifying each zero as either an isolated zero or a zerotree root. However, in the Sodagar reference, to determine whether a node is a zerotree root, a coefficient value is used from a node in a wavelet tree. In this manner, there is no determination made as to which of the bits indicate zeros. Thus, the teachings of the Zador reference clearly differs from the claim 2 limitations.

Even if combined, the Zador reference and the Sodagar reference fail to teach determining which of the bits indicate zero in a manner claimed in claim 2. The Sodagar reference merely teaches a method for encoding an input image using a wavelet transform to produce a wavelet tree. In the Sodagar reference, no determination is made as to which of the bits indicate zeros to classify each zero as either an isolated zero or a zerotree root. A wavelet tree is simply mapped on another wavelet tree and coefficients of the second wavelet tree are coded according to the method of Sodagar. While the first wavelet tree is unbalanced wavelet tree, the second wavelet tree is a balanced wavelet tree having a coefficient frame that is different from a coefficient frame from the first wavelet tree. In this manner, coding of the bits is not accomplished by determining which bits indicate zeros and classifying as each zero as either an isolated zero or a zerotree root as claimed in claim 2.

The Examiner points to column 2, lines 30-42 in the Sodagar reference, teaching both the determination and classifying limitations of claim 2. However, in column 2, the Sodagar reference merely indicates use of four symbols for entropy coding including the zerotree root and isolated zero. In other words, the specific technique of claim 2 is not taught or suggested by the Sodagar reference when coding the bits. Therefore, the combination of the Zador and Sodagar references fails to teach as a whole all the limitations of claim 2. The Applicant respectfully submits that the Section 103 rejection of claim 2 is not established as a *prima facie* case of obviousness is missing. Thus, the Section 103 rejection of claim 2 should be reversed.

C. Is claim 5 rendered obvious over the Zador reference and in view of the Ogata reference ?

Claim 5 calls for taking the difference of two successive discrete wavelet transform coded frames. The Examiner acknowledges that the Zador reference does not explicitly teach

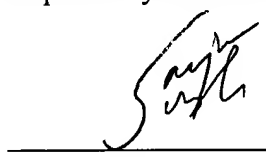
these limitations in claim 5. In rejecting claim 5, the Examiner points to teaching in the Ogata reference of Figure 2a and the 5th full paragraph in column 7. However, the Ogata reference merely teaches difference between a predicted wavelet coefficient and the wavelet coefficient to be encoded. *See,* ll. 32-34 in col. 7 of the Ogata reference.

As such, there is no teaching as to taking the difference of two successive discrete wavelet transform coded frames. Accordingly, the Section 103 rejection of claim 3 is not *prima facie* obvious to one of ordinary skill in the pertinent art. There is no motivation to combine is presented in the Zador reference since Zador reference does not teach anything even close to taking the difference of two successive discrete wavelet transformed coded frames. Even if combined, the two references could not result in the claim 5 limitations as indicated above. Thus, the Applicant respectfully requests that the Section 103 rejection of claim 5 should be reversed absent a specific teaching or a hint in the Zador and/or the Ogata references.

IX. CONCLUSION

Applicant respectfully requests that each of the final rejections be reversed and that the claims subject to this Appeal be allowed to issue.

Respectfully submitted,



Date: March 9, 2004

Senjeev K. Singh Under 37 C.F.R. § 10.9(b)
Registration No. 28,994
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Suite 100
Houston, TX 77024-1805
713/468-8880 [Phone]
713/468-8883 [Fax]

APPENDIX OF CLAIMS

1. A method comprising:

providing error data that indicate motion in an image;

representing error data as a collection of ordered bits; and

coding the bits of each order to indicate zerotree roots that are associated with the order;

performing wavelet transformations on said image with error data to provide wavelet coefficients for a wavelet transformed error image; and

in a single pass, embedding zerotree coding of the wavelet transformed error image while encoding insignificant wavelet coefficients in the course of initial passes.

2. The method of claim 1, wherein the act of coding the bits comprises:

determining which of the bits indicate zeros; and

classifying each zero as either an isolated zero or a zerotree root.

3. The method of claim 2, wherein some of the error data are descendants of some of the other error data, and wherein the act of determining comprises:

traversing a descendant tree from a bit associated with one of said some of the error data to bits associated with said other error data to locate the zerotree roots.

4. The method of claim 1 wherein providing error data includes taking the difference between two successive image representations in an image sequence.

5. The method of claim 4 wherein taking the difference includes taking the difference of two successive discrete wavelet transform coded frames.

6. The method of claim 1 including coding said bits based on whether or not the data exceeds a predetermined threshold value.

7. An article comprising a storage medium readable by a processor-based system, the storage medium storing instructions to enable a processor to:

provide error data that indicate motion in an image;

represent error data as a collection of ordered bits;

code the bits of each order to indicate zerotree roots that are associated with the order;

perform wavelet transformation on said image with error data to provide wavelet coefficients for a wavelet transformed error image; and

in a single pass, embed zerotree coding of the wavelet transformed error image while encoding insignificant wavelet coefficients in the course of initial passes.

8. The article of claim 7, the storage medium comprising instructions to enable the processor to:

determine which of the bits indicate zeros; and

classify each zero as either an isolated zero or a zerotree root.

9. The article of claim 8 wherein some of the error signals are descendants of some of the other error signals, the storage medium comprising instructions to enable the processor to:

traverse a descendant tree from a bit associated with one of said some of the error data to bits associated with said other error data to locate the zerotree roots.

10. The article of claim 7 wherein the storage medium comprising instructions to enable the processor to provide error data by taking the difference between the successive image representations in an image sequence.

11. The article of claim 10, the storage medium comprising instructions to enable the processor to take the difference of two successive discrete wavelet transform coded frames.

12. The article of claim 7, the storage medium comprising instructions to enable the processor to code the bits based on whether or not the data exceeds a predetermined threshold.

13. A system comprising:

a device to generate error frames by differencing two successive frames and to represent the error frames as a collection of ordered bits; and

an element to code the bits of each order to indicate zerotree roots that are associated with the order, perform wavelet transformation on said image with error data to

provide wavelet coefficients for a wavelet transformed error image, and in a single pass, embed zerotree coding of the wavelet transformed error image while encoding insignificant wavelet coefficients in the course of initial passes.

14. The system of claim 13 wherein said device includes a processor and a storage medium storing instructions to enable the processor to provide error data that indicate motion in the image, represent the error data as a collection of ordered bits, and encode the bits of each order to indicate zerotree roots that are associated with the order.

15. The system of claim 13 wherein said device codes said bits based on whether or not the data exceeds a predetermined threshold value.

2700 AF

TRANSMITTAL OF APPEAL BRIEF (Large Entity)

Docket No.
ITL.0210P1US

Re Application Of: **TINKU ACHARYA, PRABIR K. BISWAS & NILOY J. MITRA**

Serial No.	Filing Date	Examiner	Group Art Unit
9/723,123	NOVEMBER 27, 2000	TIMOTHY M. JOHNSON	2625

Invention:

WAVELET ZEROTREE IMAGE CODING OF ORDERED BITS

TO THE COMMISSIONER FOR PATENTS:

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on

The fee for filing this Appeal Brief is: **$330.00**

☒ A check in the amount of the fee is enclosed.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **20-1504**

Signature

Dated: **March 9, 2004**

Sanjeev K. Singh Under C.F.R. § 10.9(b)
Registration Number 28,994
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Suite 100
Houston, Texas 77024
713/468-8880 [Phone]
713/468-8883 [Facsimile]

I certify that this document and fee is being deposited on **March 9, 2004** with the U.S. Postal Service as first class mail under 37 C.F.R. 1.8 and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Signature of Person Mailing Correspondence

Rebecca R. Ginn

Typed or Printed Name of Person Mailing Correspondence

cc: